Research

FEBRUARY 23, 2017

Prophet: forecasting at scale

By: Sean J. Taylor, Ben Letham

Today Facebook is **open sourcing Prophet**, a forecasting tool available in Python and R. Forecasting is a data science task that is central to many activities within an organization. For instance, large organizations like Facebook must engage in capacity planning to efficiently allocate scarce resources and goal setting in order to measure performance relative to a baseline. Producing high quality forecasts is not an easy problem for either machines or for most analysts. We have observed two main themes in the practice of creating a variety of business forecasts:

- Completely automatic forecasting techniques can be brittle and they are often too inflexible to incorporate useful assumptions or heuristics.
- Analysts who can produce high quality forecasts are quite rare because forecasting is a specialized data science skill requiring substantial experience.

The result of these themes is that the demand for high quality forecasts often far outstrips the pace at

Meta

Research

The typical considerations that “scale” implies, computation and storage, aren’t as much of a concern for forecasting. We have found the computational and infrastructure problems of forecasting a large number of time series to be relatively straightforward — typically these fitting procedures parallelize quite easily and forecasts are not difficult to store in relational databases such as MySQL or data warehouses such as Hive.

The problems of scale we have observed in practice involve the complexity introduced by the variety of forecasting problems and building trust in a large number of forecasts once they have been produced. Prophet has been a key piece to improving Facebook’s ability to create a large number of trustworthy forecasts used for decision-making and even in product features.

Where Prophet shines

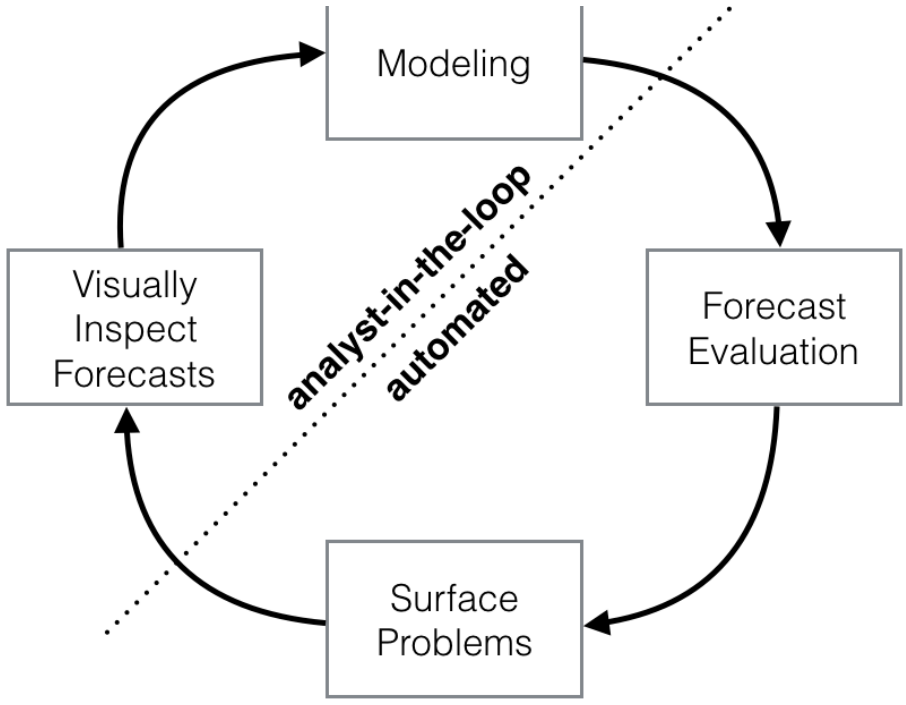
Not all forecasting problems can be solved by the same procedure. Prophet is optimized for the business forecast tasks we have encountered at Facebook, which typically have any of the following characteristics:

- hourly, daily, or weekly observations with at least a few months (preferably a year) of history
- strong multiple “human-scale” seasonalities: day of week and time of year
- important holidays that occur at irregular intervals that are known in advance (e.g. the Super Bowl)
- a reasonable number of missing observations or large outliers
- historical trend changes, for instance due to product launches or logging changes
- trends that are non-linear growth curves, where a trend hits a natural limit or saturates

We have found Prophet’s default settings to produce forecasts that are often accurate as those produced by skilled forecasters, with much less effort. With Prophet, you are not stuck with the results of a completely automatic procedure if the forecast is not satisfactory — an analyst with no training in time series methods can improve or tweak forecasts using a variety of easily-interpretable parameters. We have found that by combining automatic forecasting with analyst-in-the-loop forecasts for special cases, it is possible to cover a wide variety of business use-cases. The following diagram illustrates the forecasting process we have found to work at scale:

Meta

Research



For the modeling phase of the forecasting process, there are currently only a limited number of tools available. Rob Hyndman's excellent **forecast package in R** is probably the most popular option, and Google and Twitter have both released packages with more specific time series functionality — **CausalImpact** and **AnomalyDetection**, respectively. As far as we can tell, there are few open source software packages for forecasting in Python.

We have frequently used Prophet as a replacement for the **forecast** package in many settings because of two main advantages:

1. **Prophet makes it much more straightforward to create a reasonable, accurate forecast.** The forecast package includes many different forecasting techniques (ARIMA, exponential smoothing, etc), each with their own strengths, weaknesses, and tuning parameters. We have found that choosing the wrong model or parameters can often yield poor results, and it is unlikely that even experienced analysts can choose the correct model and parameters efficiently given this array of choices.
2. **Prophet forecasts are customizable in ways that are intuitive to non-experts.** There are smoothing parameters for seasonality that allow you to adjust how closely to fit historical cycles, as well as smoothing parameters for trends that allow you to adjust how aggressively to follow historical trend changes. For growth curves, you can manually specify "**capacities**" or the upper limit of the growth curve, allowing you to inject your own prior information about how your forecast will grow (or decline). Finally, you can specify irregular holidays to model like the dates of the Super Bowl, Thanksgiving and Black Friday.

How Prophet works

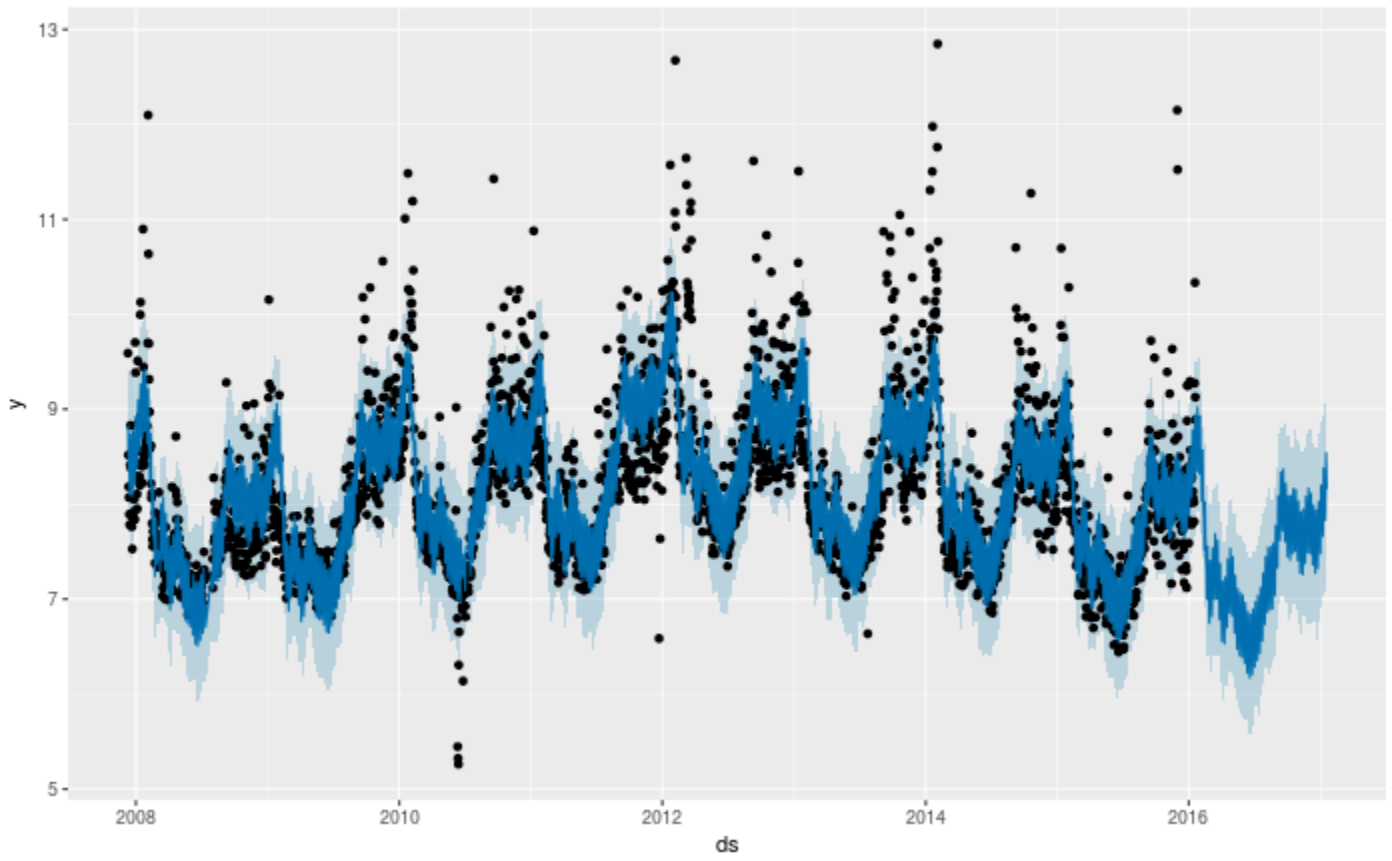
At its core, the Prophet procedure is an **additive regression model** with four main components:

- A piecewise linear or logistic growth curve trend. Prophet automatically detects changes in trends by selecting changepoints from the data.
- A yearly seasonal component modeled using Fourier series.
- A weekly seasonal component using dummy variables.
- A user-provided list of important holidays.

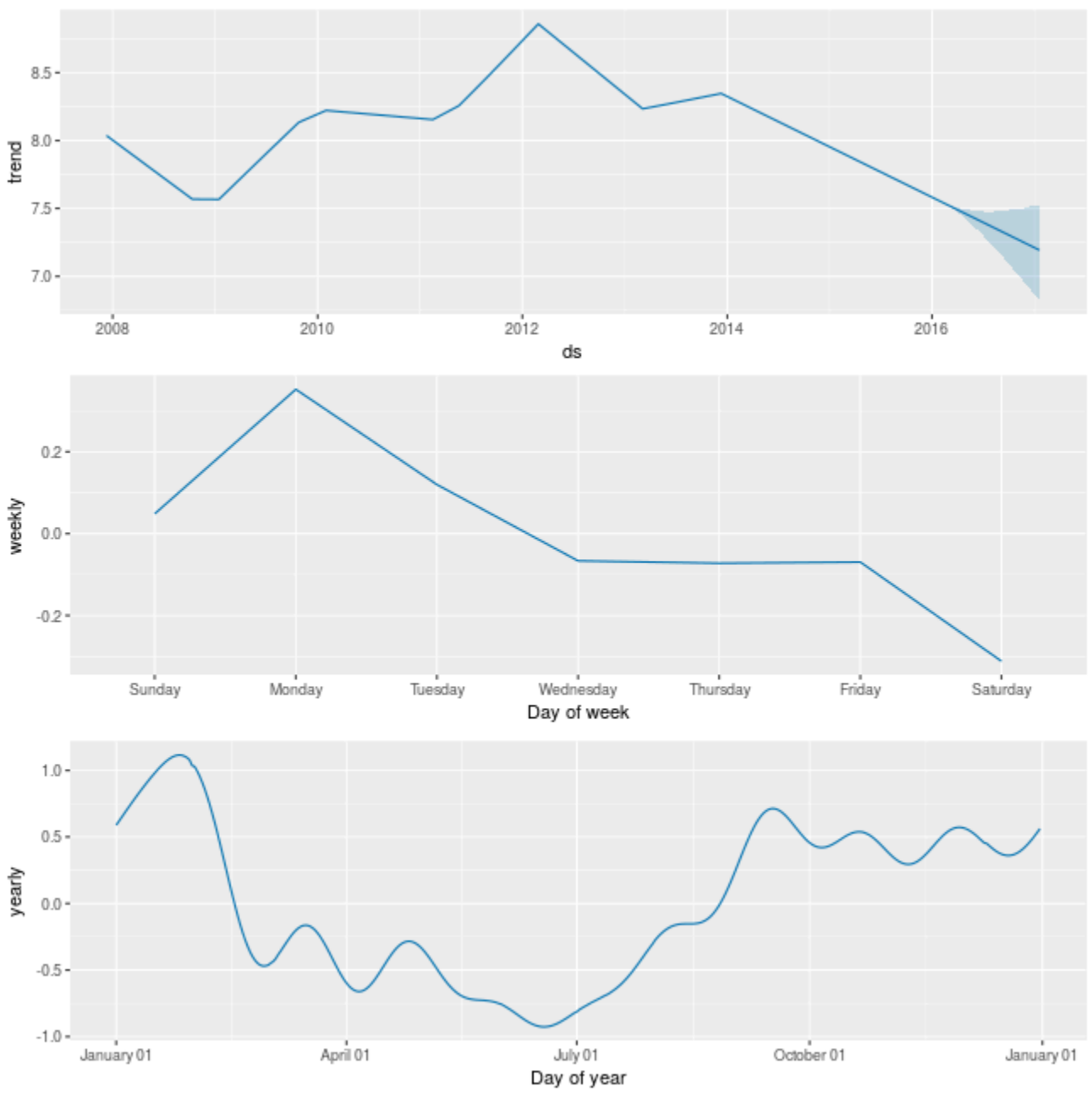
As an example, here is a characteristic forecast: log-scale page views of **Peyton Manning's Wikipedia page** that we downloaded using the **wikipediatrend** package. Since Peyton Manning is an American

Meta

Research



Prophet will provide a components plot which graphically describes the model it has fit:



This plot more clearly shows the yearly seasonality associated with browsing to Peyton Manning's page (football season and the playoffs), as well as the weekly seasonality: more visits on the day of and after

Meta

very flexible regression model (somewhat like curve-fitting) instead of a traditional time series model for this task because it gives us more modeling flexibility, makes it easier to fit the model, and handles missing data or outliers more gracefully.

By default, Prophet will provide uncertainty intervals for the trend component by simulating future trend changes to your time series. If you wish to model uncertainty about future seasonality or holiday effects, you can run a few hundred HMC iterations (which takes a few minutes) and your forecasts will include seasonal uncertainty estimates.

We fit the Prophet model using **Stan**, and have implemented the core of the Prophet procedure in Stan's probabilistic programming language. Stan performs the MAP optimization for parameters extremely quickly (<1 second), gives us the option to estimate parameter uncertainty using the **Hamiltonian Monte Carlo** algorithm, and allows us to re-use the fitting procedure across multiple interface languages. Currently we provide implementations of Prophet in both Python and R. They have exactly the same features and by providing both implementations we hope to make our forecasting approach more broadly useful in the data science communities.

How to use Prophet

The simplest way to use Prophet is to install the package from PyPI (Python) or CRAN (R). You can read our **quick start guide** and dive into our **comprehensive documentation**. If you're looking for a fun source of time series data, we recommend trying the **wikipediatrend** package which will download historical page views on Wikipedia pages.

Help us improve Prophet

There are two main ways to help us improve Prophet. First, you can try it yourself and tell us about your results. We're always looking for more use cases in order to understand when Prophet performs well and when it does not. Second, there are plenty of features that are left to build! **We welcome pull requests** with bugfixes and new features. Check out **how to contribute**, we look forward to engaging with the community to make Prophet even more broadly useful.

Areas ECONOMICS & COMPUTATION DATA SCIENCE SYSTEMS & INFRASTRUCTURE

Tags CORE DATA SCIENCE

Share

Research

February 7, 2023

Improving Meta's global maps

:ngineering at Meta

February 2, 2023

The future of time-series forecasting, with RFP winner B. Aditya Prakash

December 12, 2022

Efficient Multi-Objective Neural Architecture Search with Ax

'yTorch blog

Meta

Research

Meta

Meta Research

Publications

Requests for Proposals

RFP Funding Recipients

Fellows

Datasets

Careers

Visit Our Other Blogs

Blog

Engineering

Meta AI

Meta Quest

Tech at Meta

earch